活用TIPs

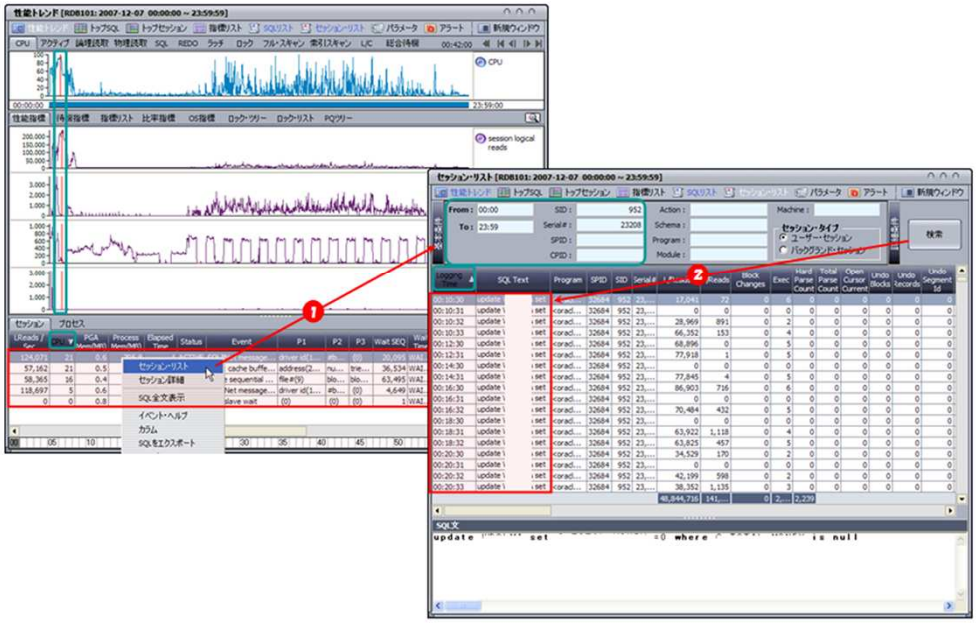
活用TIPsリスト

■ 開発
- AP処理(SQL)フローの追跡
- 長文のSQLを見やすくしたい(SQLの標準化)

■ ラッシュテスト
- 特定時間帯の上位SQLを確認したい
- 特定時間帯で特定SQLの実施統計を確認したい
- 高負荷のFULL TABLE SCANを行っているSQLを特定したい
- ユーザー定義情報を時系列で監視
- ユーザー定義情報のスナップショット取得
- ロックの発生状況を確認したい
- 特定時間帯の性能測定

■ 運用
- 性能低下階層の切り分け:DB orその他？
- 突然性能低下した、何から調べる？
- サポート依頼に必要なデータの抽出
- 接続数の変動を確認したい
- 接続エラーの回避
- CPU過負荷時の調査手順
- 「ORA-4031」対処 → リテラルSQL確認
- 過去の特定時刻実行計画を確認
- 実行計画が変わったSQLの確認
- SQLがアクセスしたオブジェクトを確認したい

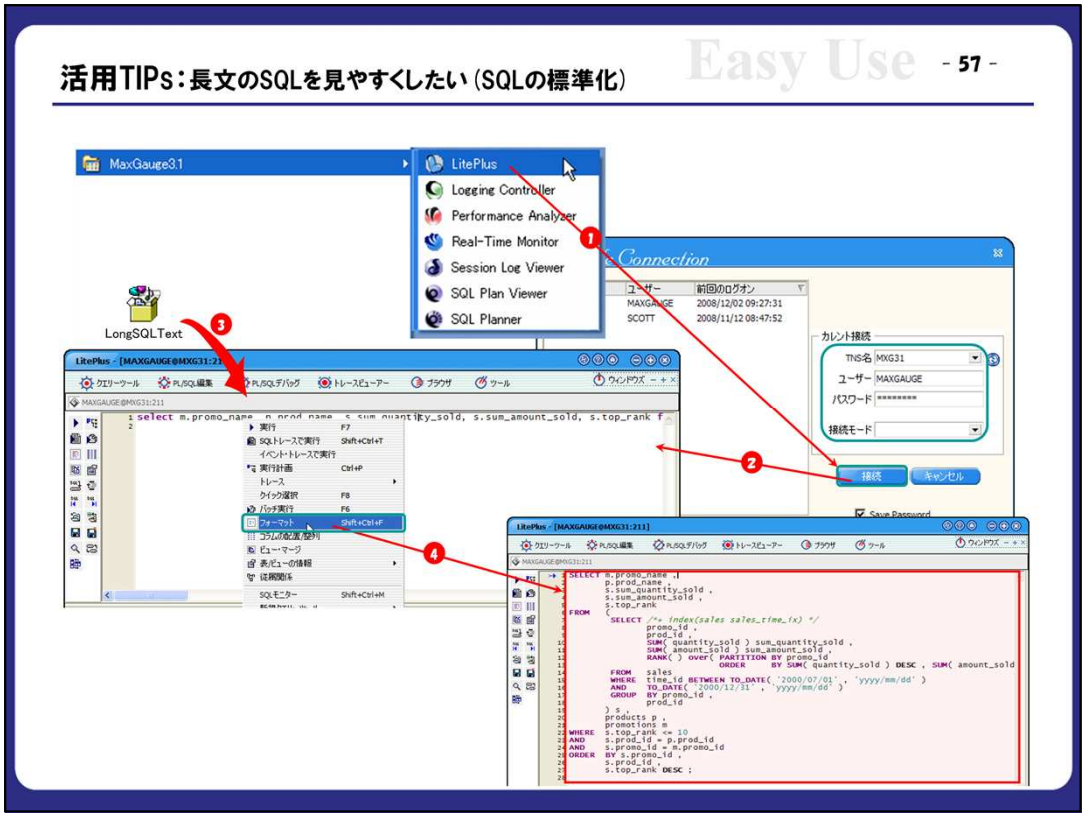
活用TIPs：AP処理（SQL）フローの追跡



①Performance Analyzerの「セッション」タブから分析すべきセッションを指定し、右クリックの「セッション・リスト」を選択します。

→ 例ではCPU使用率の高いセッションを選択しました。

②「検索」行って、「Logging Time」の逆順でソートし、1秒単位の実行SQLの流れを確認します。

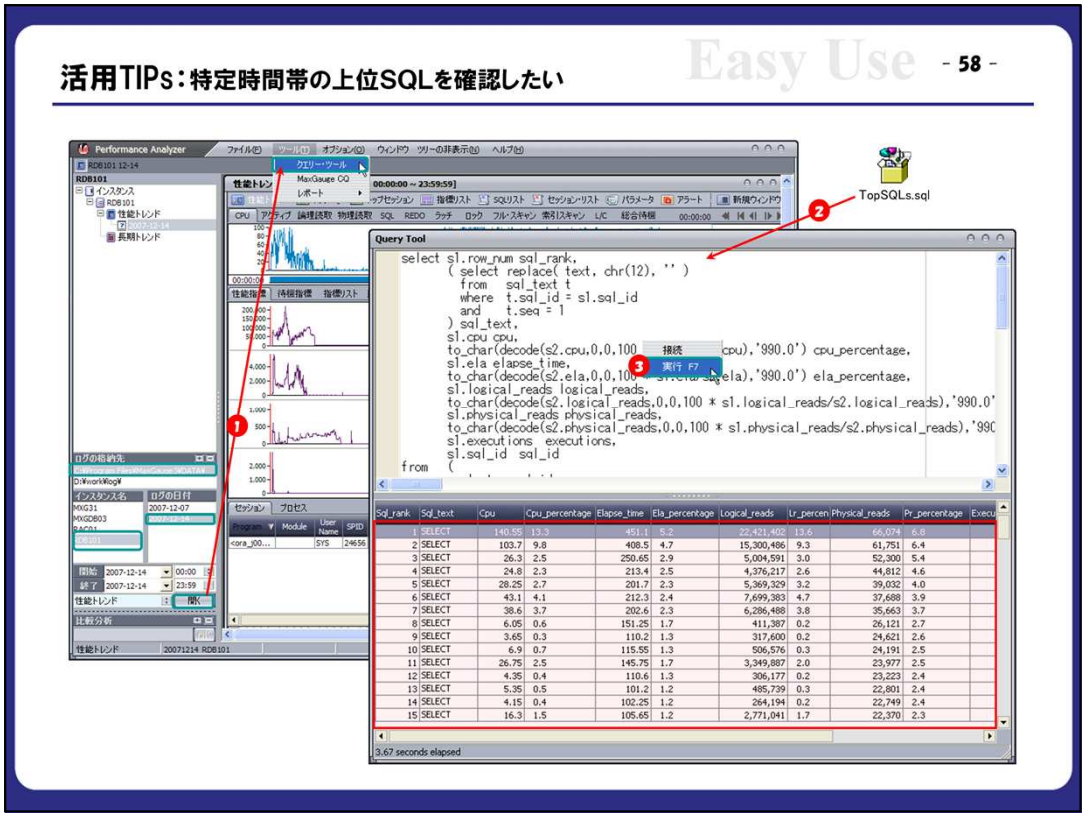


①「スタート」より、MaxGaugeのLitePlusを選択します。

②DBに接続するため、適切な「TNS名、ユーザー、パスワード」を入力し、「確認」を押します。

③長文のSQLテキストをLitePlusのコマンド領域にコピーします。

④右クリックの「フォーマット」メニューをクリックします。
→ 標準編集されたSQLテキストを確認し、コピーなどで活用します。



①Performance Analyzerで、確認対象の「インスタンス名」、「ログの日付」を選択して、「クエリー・ツール」を開きます。

②「TopSQLs」テキストを開いて、確認する上位SQL(以下)から対象時間帯「from_time >= '00:00:00' AND to_time <= '23:59:59'」を修正し、コマンドスペースに貼り付けます。

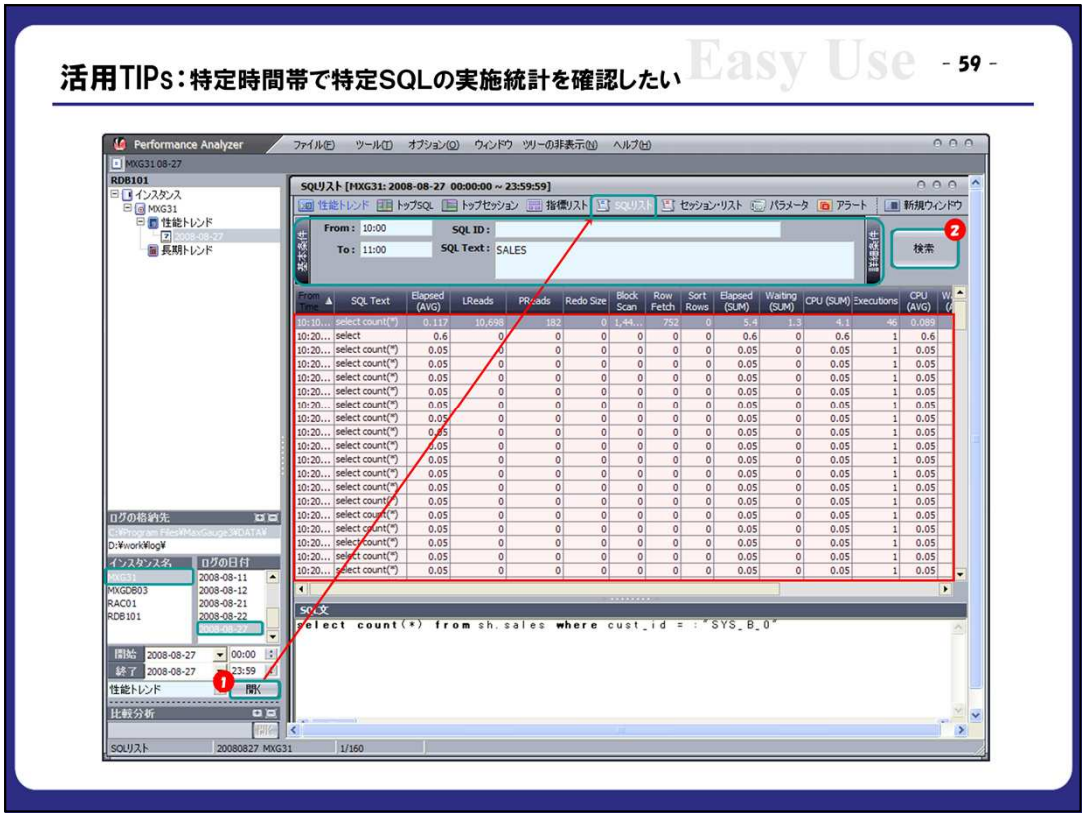
- Top SQLs by CPU
- Top SQLs by 実行時間
- Top SQLs by 論理読取
- Top SQLs by 物理読取
- Top SQLs by 実行回数

③「F7」あるいは右クリックの「実行」メニューで、「②」のSQLを実行します。

→ 例の結果は、「 Top SQLs by CPU 」の「from_time >= '14:00:00' AND to_time <= '15:00:00'」で取得しました。

→ 各上位SQLと、全SQLに対する割合(負荷率)を確認します。

※ ログファイルのDB構成の詳細は、マニュアル「volume II_jpn_3.1.pdf → 参考. Oracle Database Liteのテーブル情報」をご参照ください。

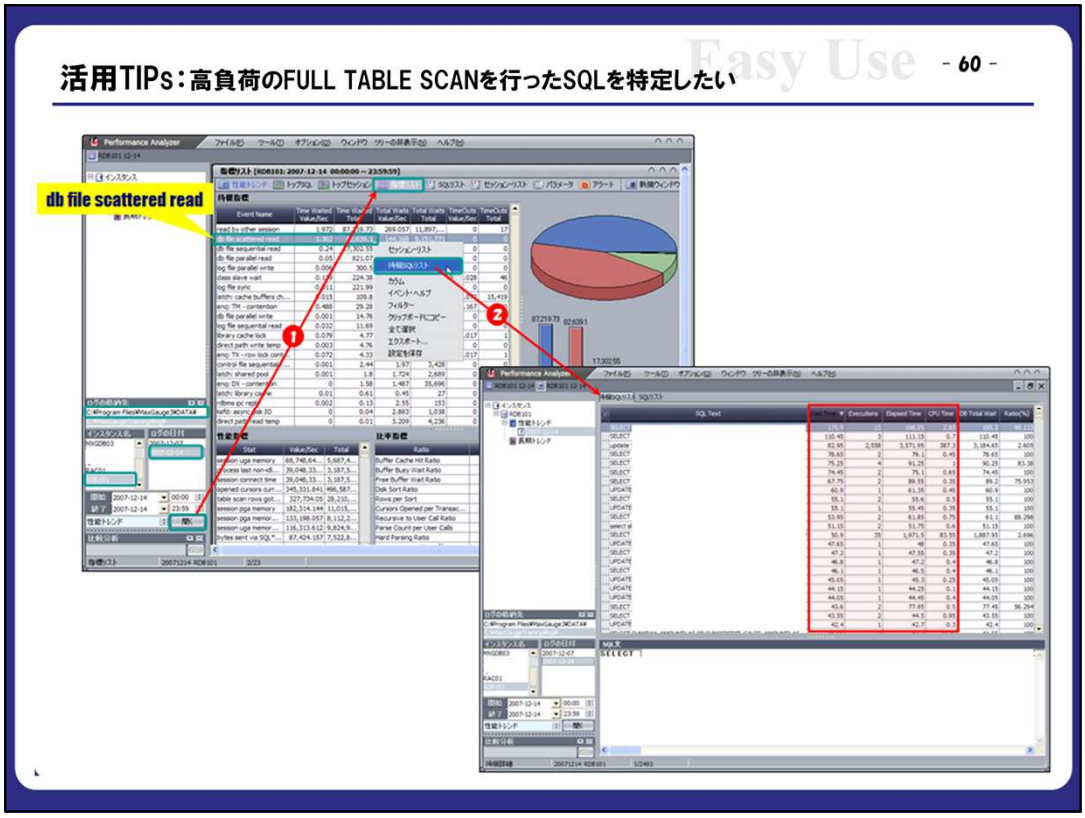


①Performance Analyzerで、確認対象の「インスタンス名」、「ログの日付」を選択して、「SQLリスト」タブを開きます。

②「基本条件」、「詳細条件」に適切な条件を指定して、「検索」を押します。

→ 例は、「10:00～11:00」間で、「SALES」表にアクセスしたSQLを抽出しました。

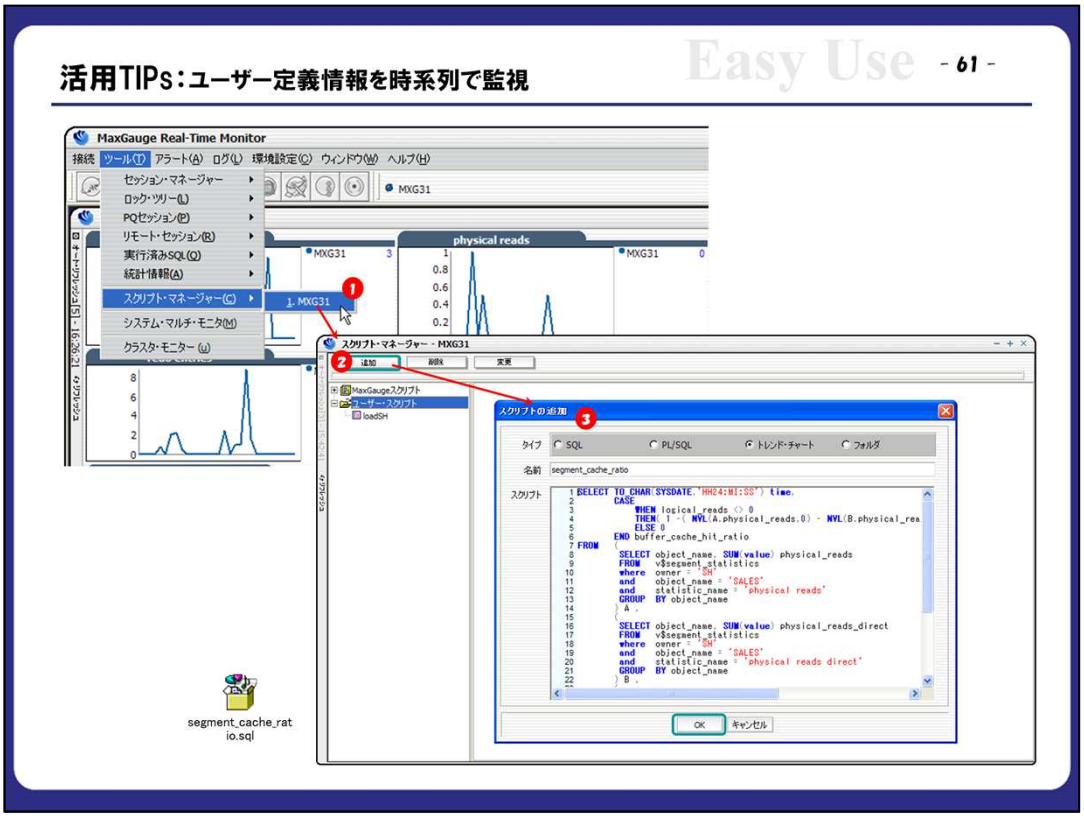
「詳細条件」では、「Elapsed Time >= 100」、「CPU >= 60」など実行統計に対しても条件指定が出来ます。



①Performance Analyzerで、確認対象の「インスタンス名」、「ログの日付」を選択します。

②「待機リスト」の「db file scattered read」を選択して、右クリックの「待機SQLリスト」を選択します。

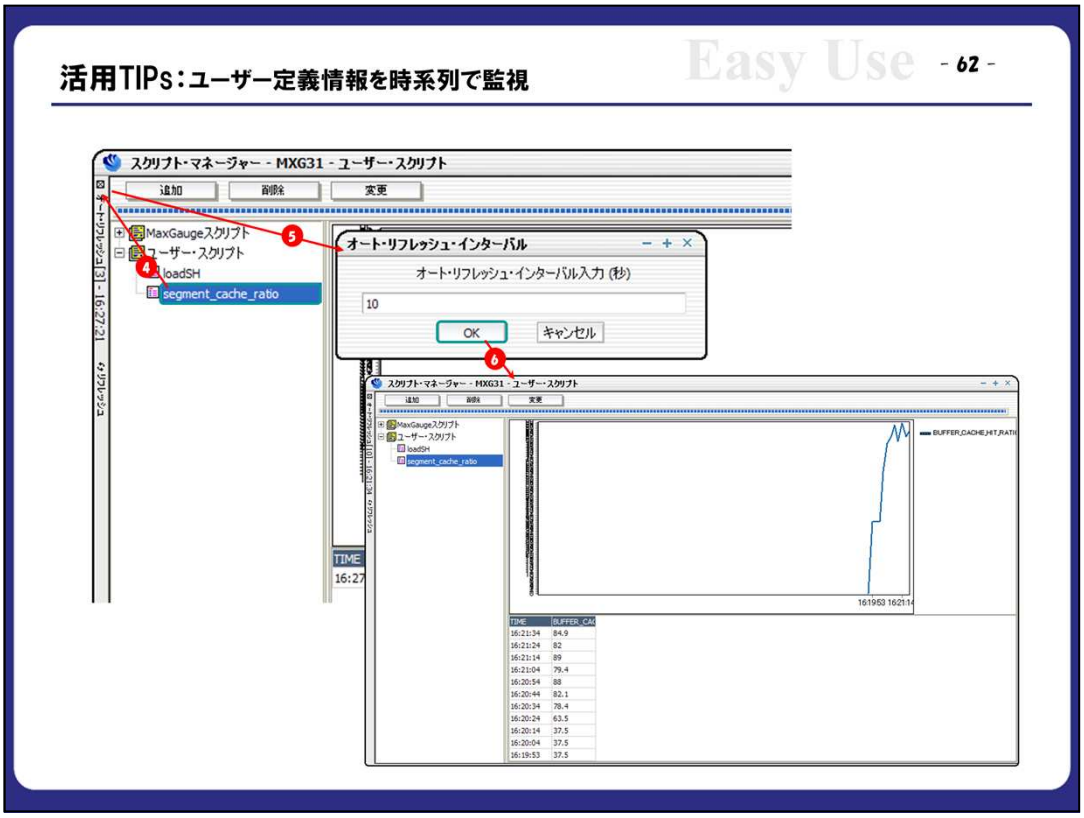
→「Wait Time」、「Elapsed Time」、「CPU Time」の項目を基準にソートして、上位SQLを確認します。



①Real-Time Monitorの「スクリプト・マネージャー」から該当インスタンスを選択します。

②「ユーザー・スクリプト」を選択し、「追加」を押します。

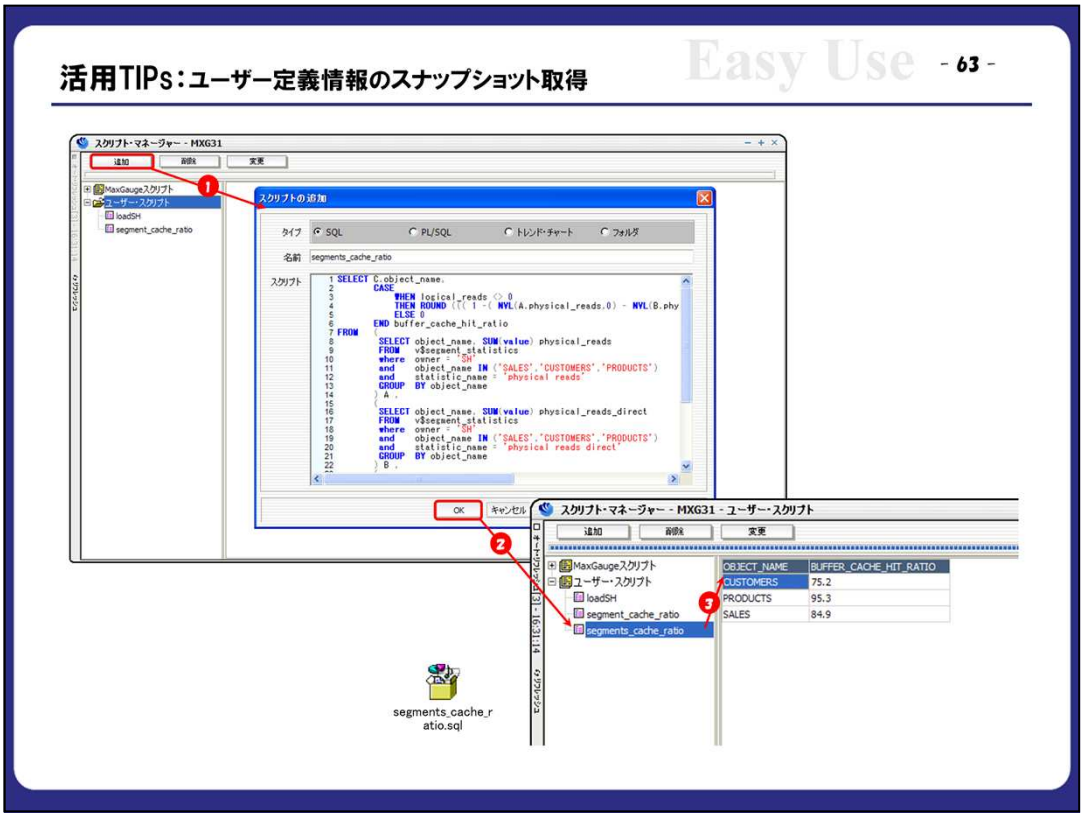
③「トレンド・チャート」を選択、名前とスクリプトを入力し、「OK」をクリックします。
但し、スクリプトは「time」を含み、単一のレコードを返すように作成します。



④対象の定義スクリプトを実行します。

⑤「オート・リフレッシュ」をダブルクリックします。

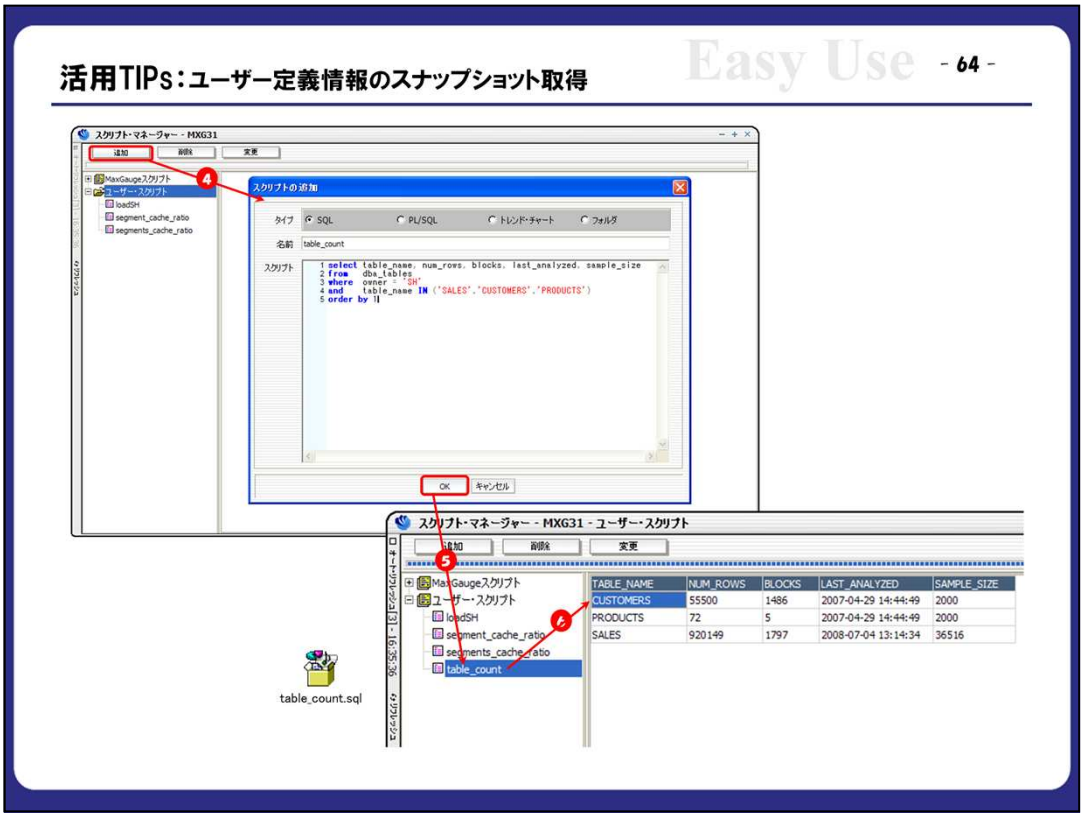
⑥適切なインターバルを入力し、「OK」を押して、トレンド及び数値をモニタリングします。



①「スクリプト・マネージャー」の「ユーザー・スクリプト」を選択し、「追加」を押します。

②「SQL」あるいは「PL/SQL」を選択して、名前とスクリプトを入力し、「OK」をクリックします。

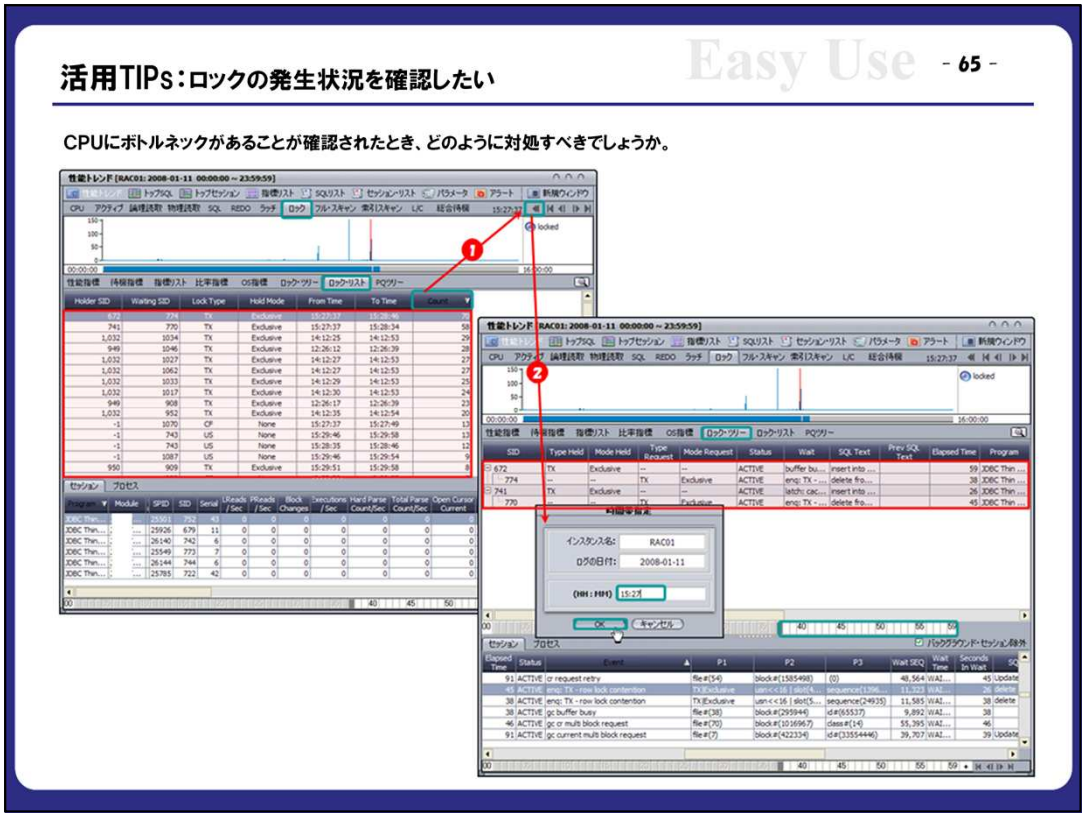
③「②」のツリー名をダブルクリックで、実行します。



④「スクリプト・マネージャー」の「ユーザー・スクリプト」を選択し、「追加」を押します。

⑤「SQL」あるいは「PL/SQL」を選択して、名前とスクリプトを入力し、「OK」をクリックします。

⑥「②」のツリー名をダブルクリックで、実行します



①Performance Analyzerで、「ロックリスト」タブの「Count」項目を逆順でソートし終日発生しましたロックリストを確認します。

→ 最長ロック:「774」セッションが、「15:27:37～15:28:46」の「70」秒間、「672」セッション保持の「Exclusive」モードの「 TX 」ロックを待機しました。

②「①」のリストで確認すべきロックの発生時刻を指定して、関連セッション、SQLのロックツリーを確認します。

活用Tips：特定時間帯の性能測定

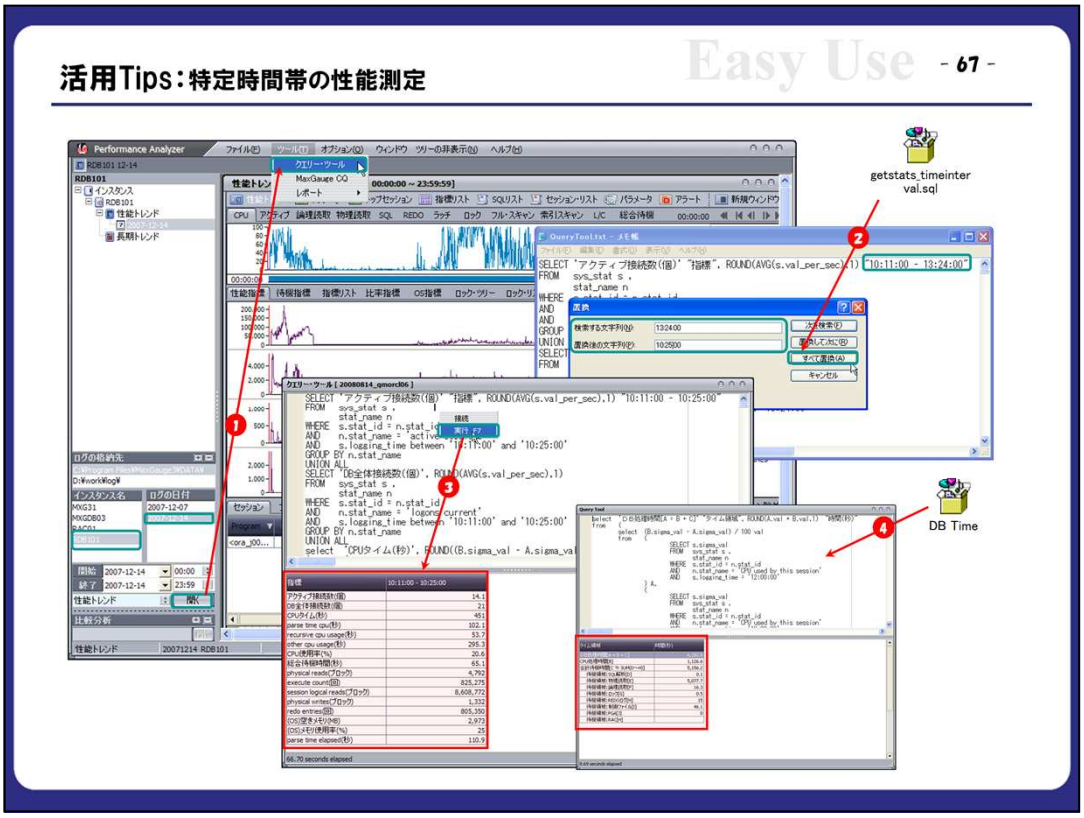
APの性能測定時に、データベース処理の性能はどの項目をどのように測定すればよいでしょうか。また、過去のある時間帯のデータベース処理の性能の概要を把握するためにはどの項目をどのように確認すればよいでしょうか。

右側の表はあるラッシュテストの結果を纏めたものですが、同様のデータを収集することで特定時間帯のデータベース処理の概要の把握や比較分析も簡単に行えます。

尚、パラメータ変更、CPU追加、ディスク装置の交換などデータベース環境変更の際に、その前後の同じ時間帯の統計数値を比較することで、システム運用状況の変化が一目瞭然に見えてきます。

※ ラッシュテスト結果表

区分	診断領域		2007/11/16 (14:52～15:03)	2008/06/24 (10:09～10:24)	2008/06/27 (13:25～13:44)
総合指標	接続数	active sessions(個)	16.10	15.00	15.90
		logons current(個)	21.40	18.69	19.95
	滞留	総合待機時間(秒)	451.90	83.30	148.70
	CPU	CPU タイム(秒)	836.81	1,428.98	2,350.35
		CPU Usage(%)	68.37	67.87	74.33
領域指標	作業量	execute count(回)	752,611.00	838,294.00	818,382.00
		physical reads(ブロック)	3,226,141.00	74.00	87.00
		physical writes(ブロック)	27,020.00	1,593.00	3,938.00
		session logical reads(ブロック)	11,487,160.00	7,492,340.00	7,461,532.00
		redo entries	833,497.00	793,236.00	780,197.00
	メモリ(OS)	(OS)free mem size(MB)	1,874.33	1,377.00	1,274.00
		(OS)Memory Usage(%)	55.26	66.00	69.00
	SQL解析処理	所要時間(秒)	140.44	494.58	958.29
		parse time cpu(秒)	124.49	386.91	852.04
		parse count (total)(回)	762,053.00	846,774.00	827,073.00
		parse count (hard)(回)	11,382.00	233,067.00	225,430.00
		主要待機時間(秒)	0.10	24.30	64.00
	I/O	主要待機時間(秒)	408.50	0.30	0.20
	バッファー・キャッシュ	主要待機時間(秒)	5.30	2.40	3.00
		Buffer Cache Hit Ratio(%)	69.26	100.00	100.00
	REDO	主要待機時間(秒)	31.80	30.20	46.80
		user commits(回)	11,375.00	13,005.00	12,713.00
		user rollbacks(回)	1.00	0.00	1.00
	ロック	主要待機時間(秒)	0.80	0.70	3.30
総合待機時間 + CPU タイム(秒)			1,287.61	1,513.28	2,499.05

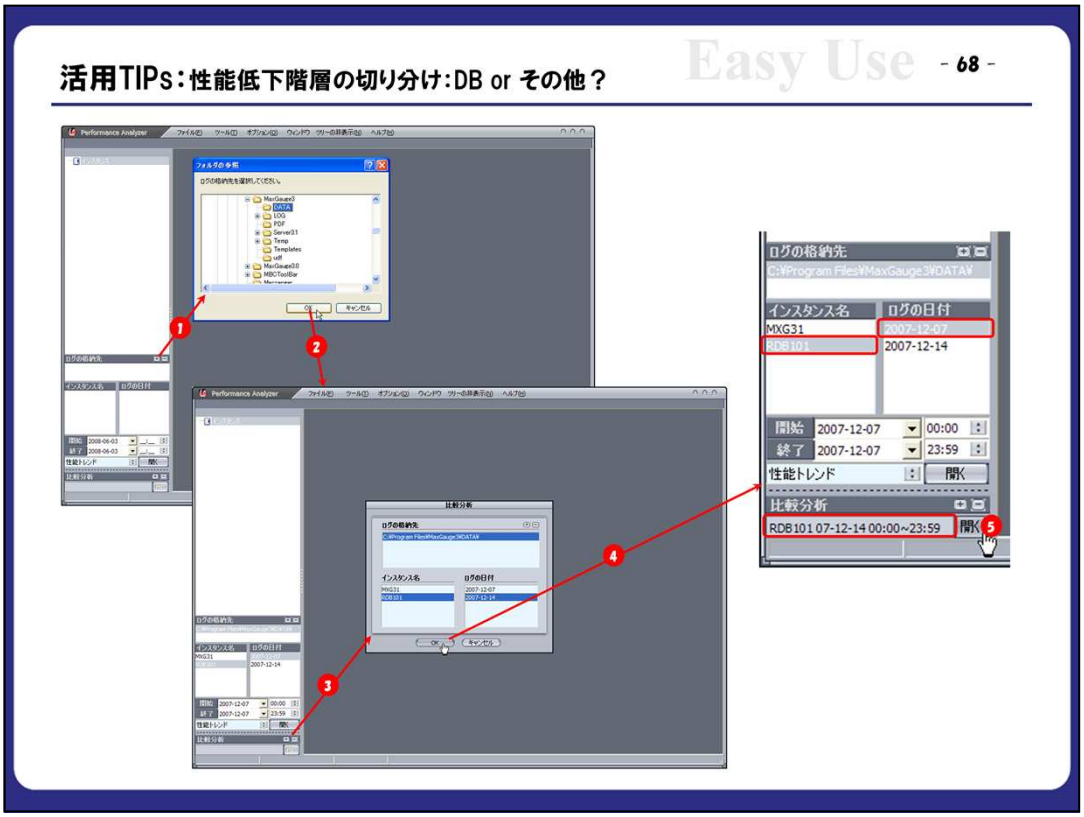


①Performance Analyzerで、確認対象の「インスタンス名」、「ログの日付」を選択して、「クエリー・ツール」を開きます。

②「getstats_timeinterval.sql 」を開いて、性能測定時間帯の開始時刻と終了時刻を設定し、コマンドスペースに貼り付けます。

③「F7」あるいは右クリックの「実行」メニューで、「②」のSQLを実行（数分所要）します。
→ 集計結果をコピーまたはエクスポートして利用することも有効です。

④DBタイムのサマリーを確認することで、該当時間帯のボトルネックがより簡単に把握できます。



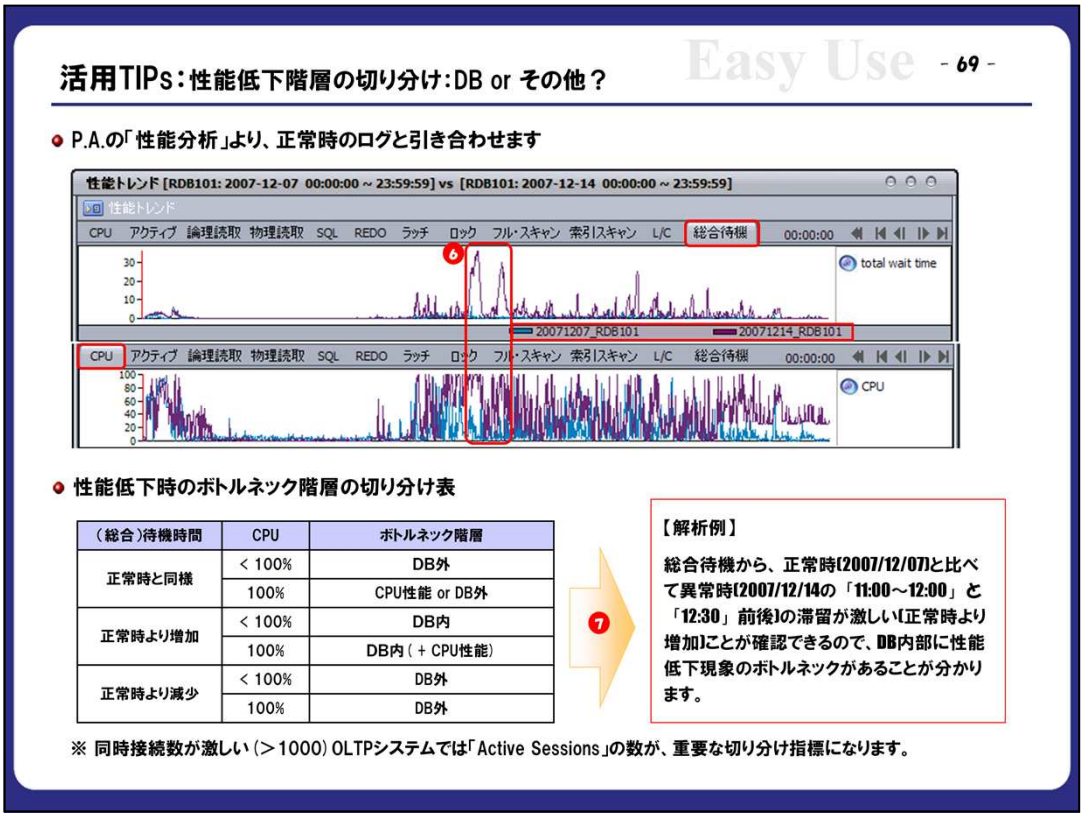
①比較元になる正常時のログの格納先の指定画面を開きます。

②適切な格納場所を指定します。
　デフォルトの格納先は「C:¥Program Files¥MaxGauge3¥DATA」になります。

③比較対象になるログの選択画面を開きます。

④性能低下現象が発生しているログの格納場所、インスタンス、日付を選択します。

⑤比較対象として選択されたログと日付を確認した上、性能比較画面を開きます。



⑥「CPU」、「総合待機」タブのトレンドを比較します。

・同一インスタンス、「12/7, 12/14」両日の２４時間のトレンドです。

・青色のグラフがRDB101インスタンスの「12/7」のトレンドです。

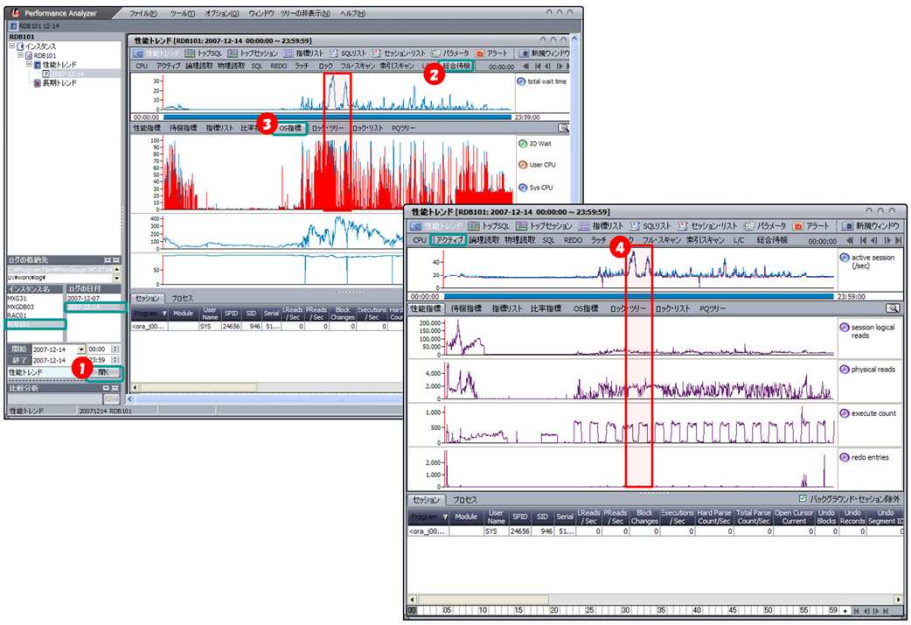
・紫色のグラフがRDB101インスタンスの「12/14」のトレンドです。

・総合待機時間が、終日「12/7」より「12/14」の滞留が多いですが、特に「11:00～12:00」時間帯と「12:30」前後の滞留現象が激しいです。

・同じ時間帯のCPU使用率が100%近く続いて、CPUリソースにも相当のロードが掛かっています。

⑦切り分け表から、DB内部に性能低下現象の原因があることが分かります。

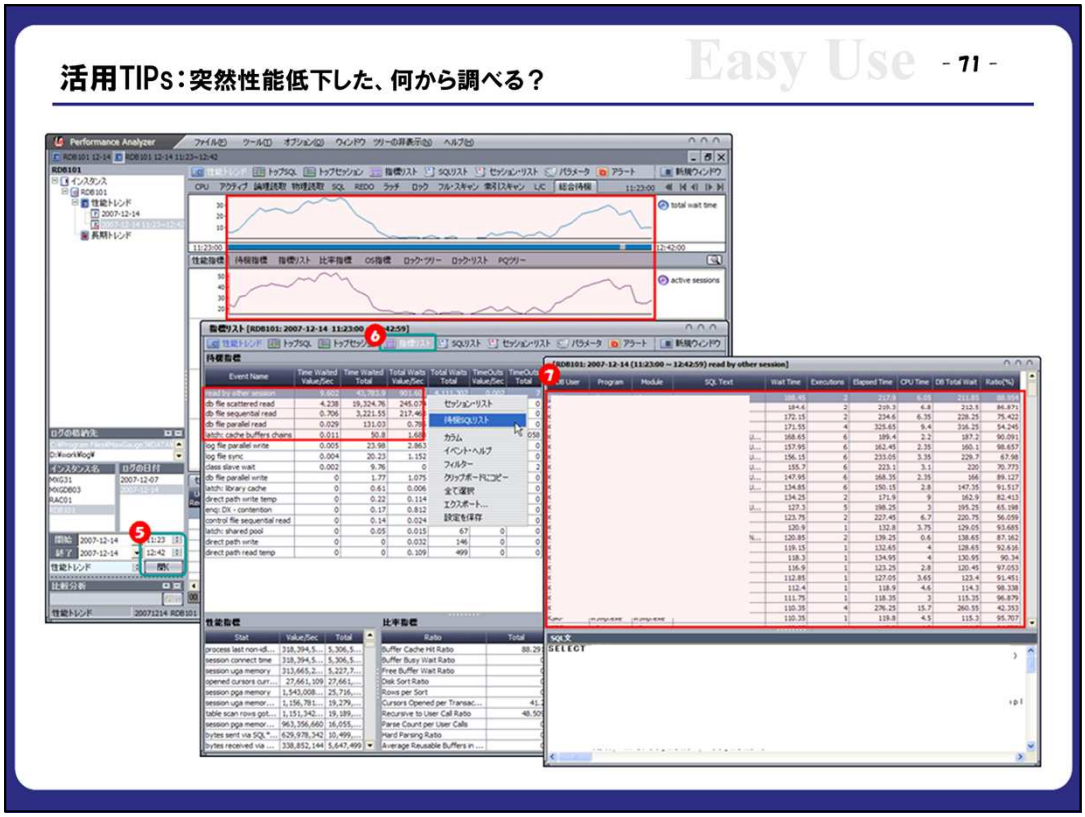
活用TIPs：突然性能低下した、何から調べる？



①Performance Analyzerから、適切な「インスタンス名」、「ログの日付」を選択して、開きます。

②「総合待機」タブで、待機時間のトレンドを確認します。
→「11:23～12:42」間の滞留が激しいことが分かります。

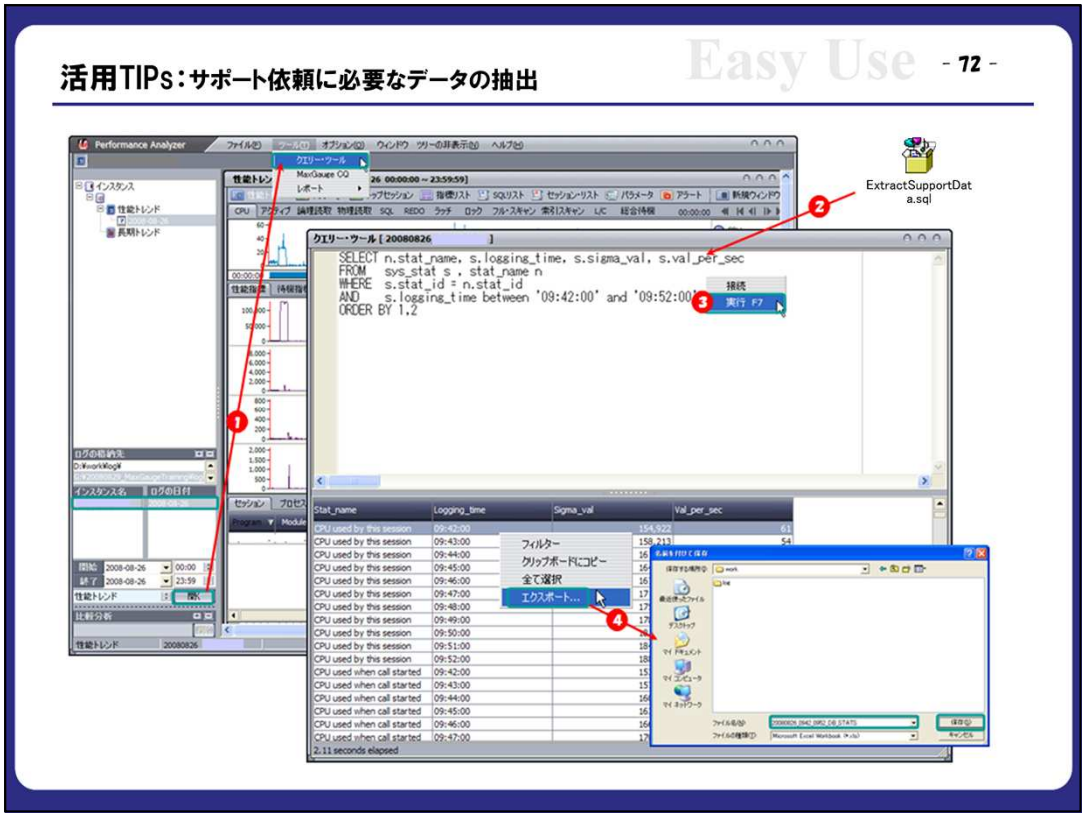
③「OS指標」タブで、「CPU」使用率のトレンドを確認します。



④「②」で確認した、時間帯を指定して拡張します。

⑤「指標リスト」の「待機指標」のトップ5を確認します。

⑥各指標の滞留関連のSQLを確認します。



①Performance Analyzerで、確認対象の「インスタンス名」、「ログの日付」を選択して、「クエリー・ツール」を開きます。

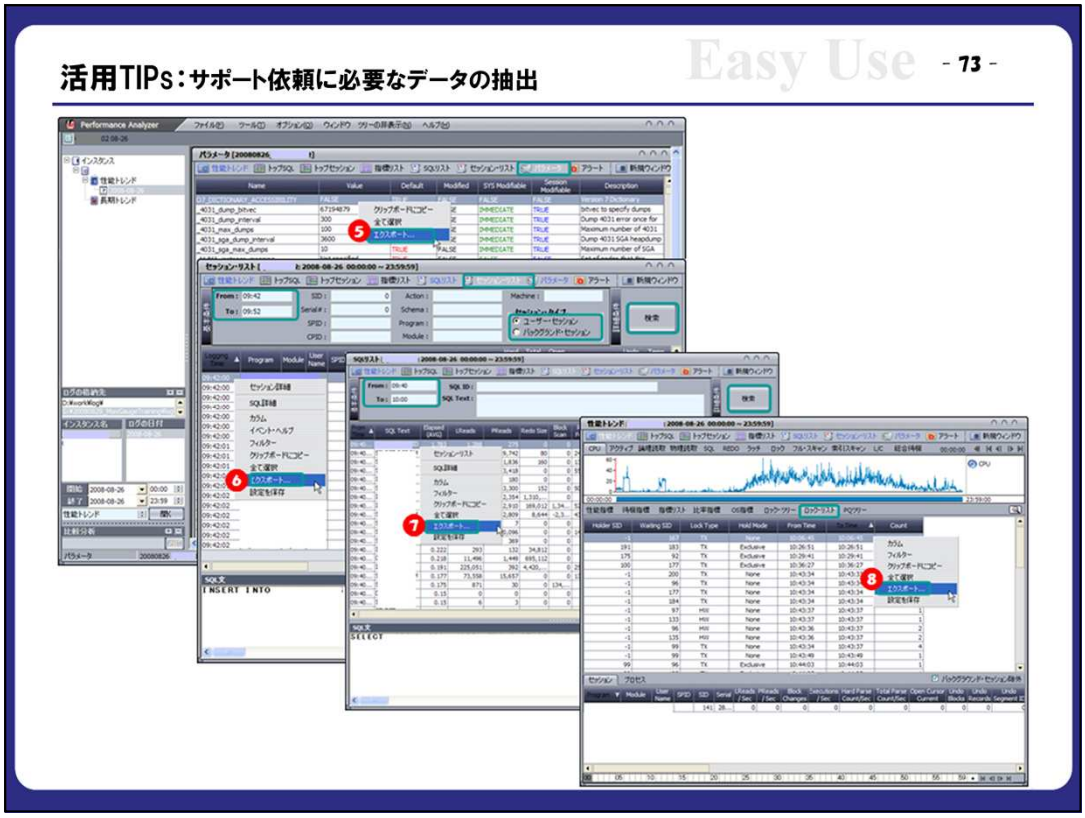
②「ExtractSupportData.sql」テキストを開いて、抽出するデータ種類のSQLの対象時間帯「s.logging_time between '09:42:00' and '09:52:00'」を修正し、コマンドスペースに貼り付けます。

→ 例は、「Oracleデータベースの性能統計(性能指標)のデータ」を貼り付けています。

③「F7」あるいは右クリックの「実行」メニューで、「②」のSQLを実行します。

④実行結果リストの右クリックメニュー「エクスポート」で、結果を保存します。

→ 同様の手順で、OS情報のデータ、Oracleデータベースのイベント(待機指標)のデータ、上位OSプロセスのデータなどを抽出します。



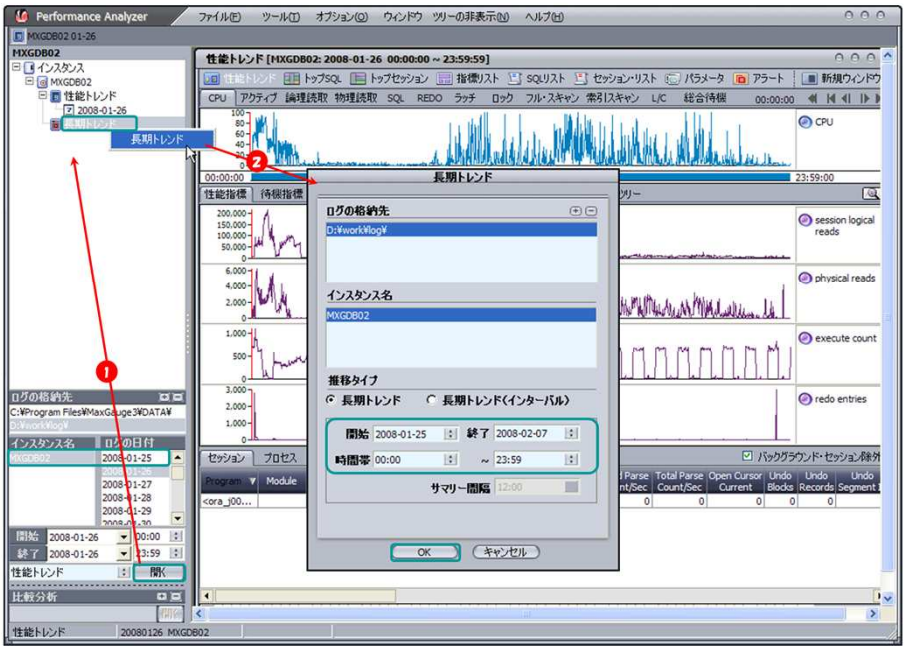
⑤「パラメータ」リストの右クリックメニュー「エクスポート」で、結果を保存します。

⑥「セッションリスト」で該当時間(From, To)指定した検索を行い、「Logging Time」順の結果リストの右クリックメニュー「エクスポート」で、結果を保存します。
→「ユーザーセッション」、「バックグラウンドセッション」両方抽出することも有効です。

⑦「SQLリスト」で該当時間(From, To)指定した検索を行い、時間順の結果リストの右クリックメニュー「エクスポート」で、結果を保存します。

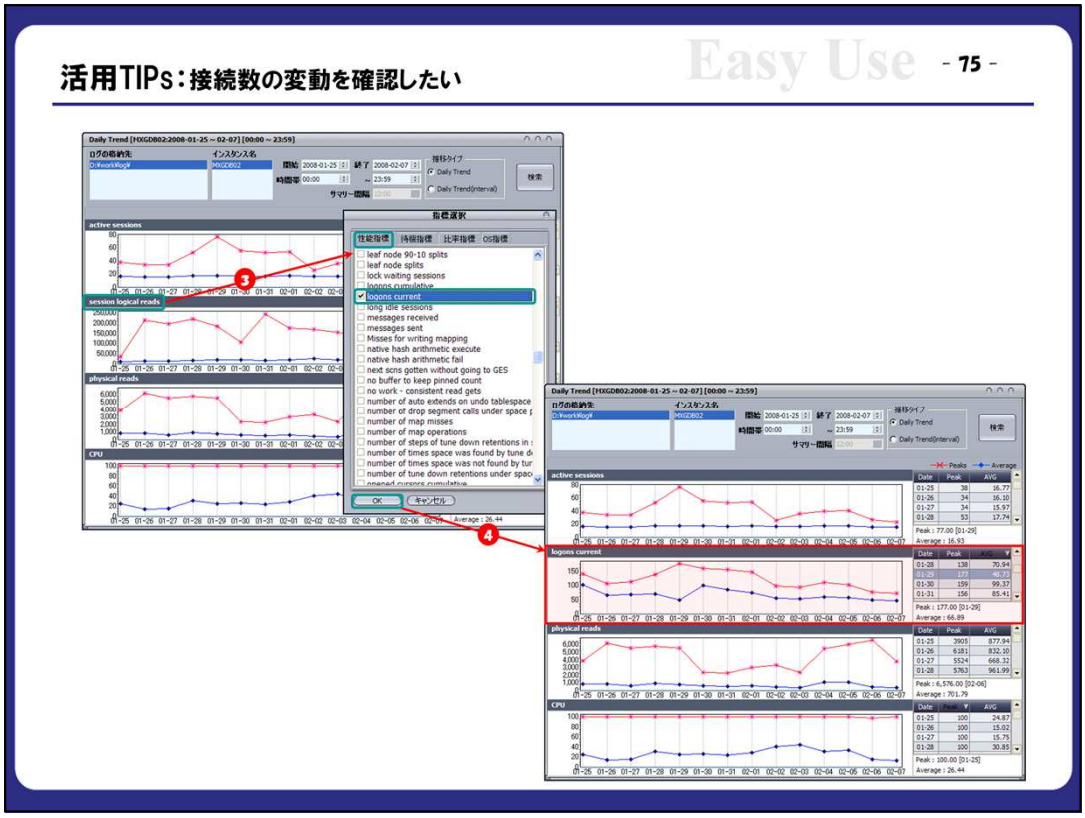
⑧「ロックリスト」の右クリックメニュー「エクスポート」で、結果を保存します。

活用TIPs：接続数の変動を確認したい



①Performance Analyzerから、適切な「インスタンス名」、「ログの日付」を選択して、開きます。
　ここでは、該当インスタンスの長期トレンドを見ますので、「インスタンス名」のみ正しければ結構です。

②ツリーの「長期トレンド」の右クリックメニューで、長期トレンドの詳細指定画面を開いて、
　適切な期間(「開始」と「終了」)と、「時間帯」を指定して、「OK」を押します。
　（期間中の対象のログ数が多いほど、表示に時間が掛かります）

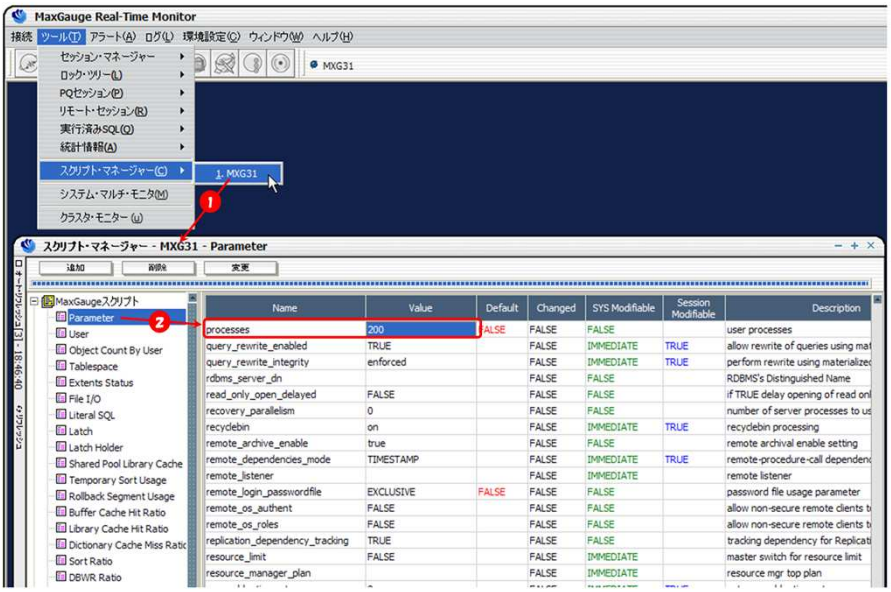


③表示された長期トレンドの指標名をダブルクリックして、「指標選択」画面を開きます。

④「性能指標」の「logons current」指標を選択し、「OK」を押します。

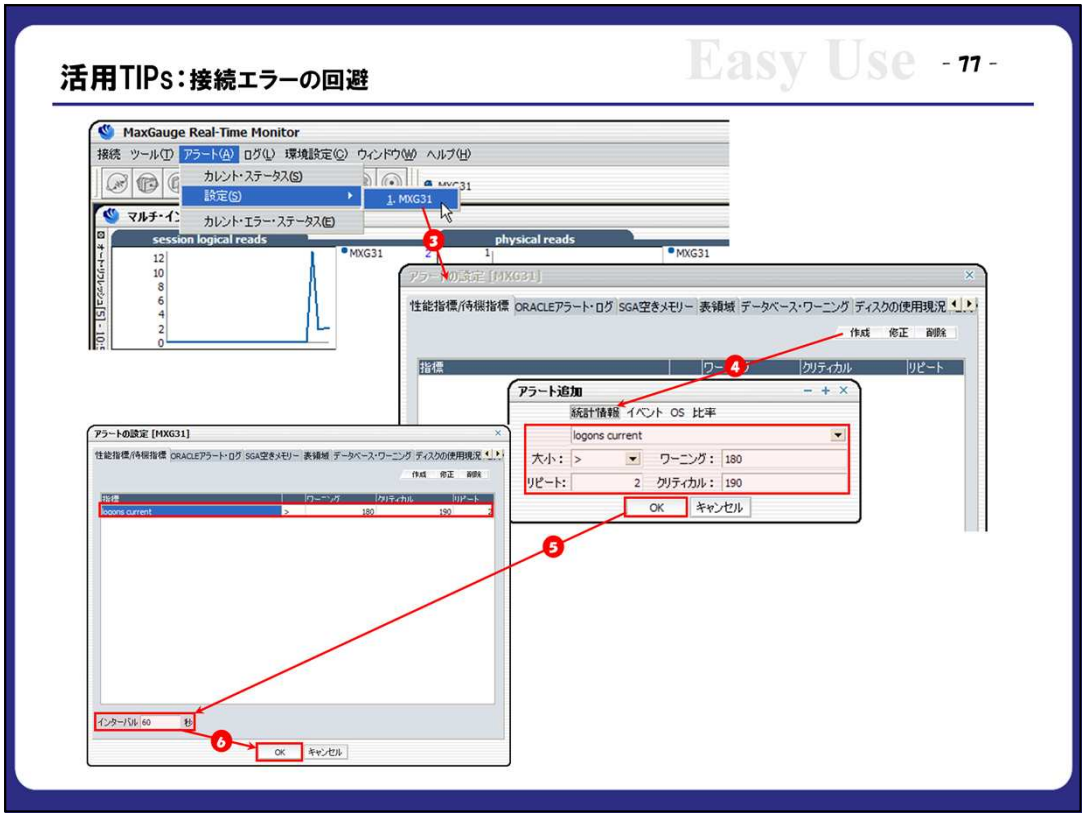
→ 指定された時間帯の日別単位に、最大接続数と平均接続数の推移グラフと表を確認します。

活用TIPs：接続エラーの回避



①Real-Time Monitorの「スクリプト・マネージャー」から該当インスタンスを選択します。

②「Parameter」をダブルクリックで実行し、その結果リストから最大プロセス数「processes」を確認します。



③Real-Time Monitorの「アラート→設定」メニューから該当インスタンスを選択します。

④「性能指標/待機指標」タブを選択して「作成」を押します。

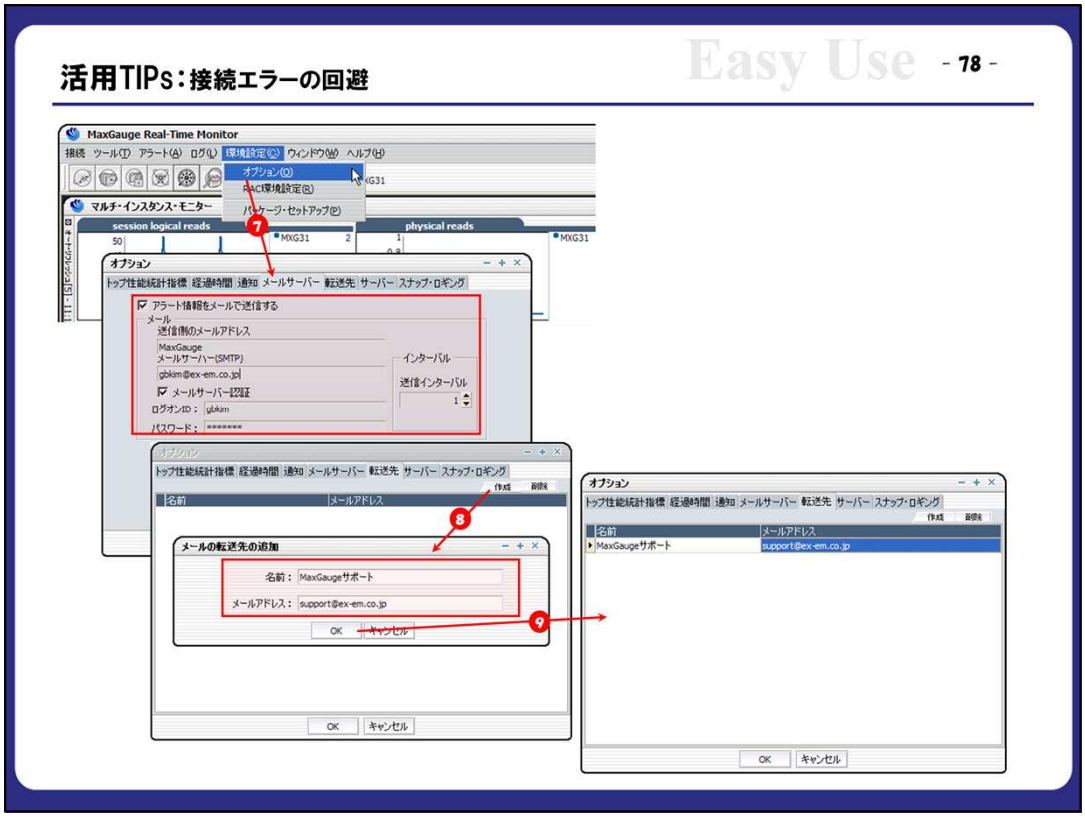
⑤「統計情報」の「logons current」指標を選択して、以下の値(例)を選択・設定し、「OK」を押します。

・大小「>」
・リピート「2」
・ワーニングの閾値「 180」
・クリティカラの閾値「190」

⑥「インターバル=60(秒)」を設定し、「OK」を押します。

→ 接続数が「180」以上「190」未満で、60秒間隔で2回連続でヒットした場合、ワーニングのアラートとを出す

接続数が「190」以上で、 60秒間隔で2回連続でヒットした場合、クリティカルのアラートとを出す



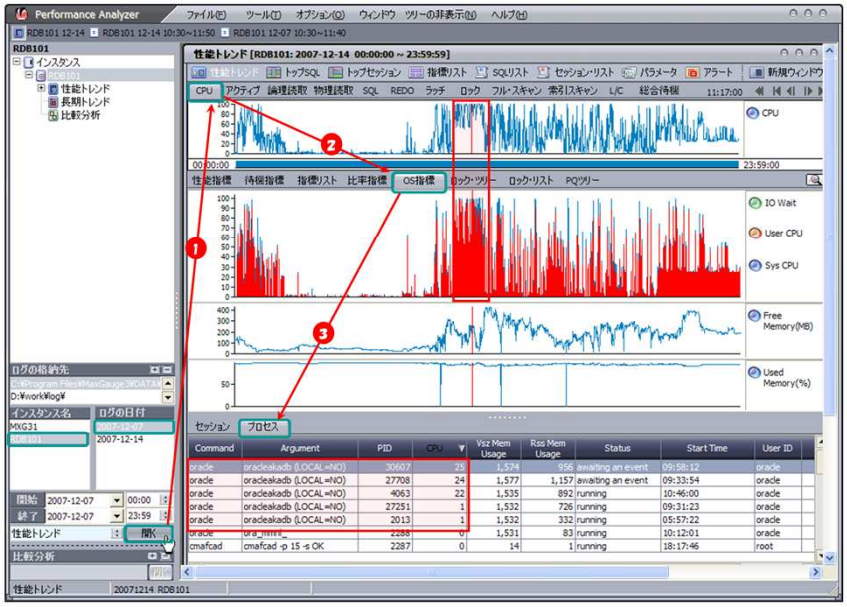
⑦Real-Time Monitorの「環境設定 → オプション」メニューを選択し、メールサーバーを設定します。

⑧「転送先」タブで「作成」を押します。

⑨転送先のメールアドレスを設定します。

活用TIPs：CPU過負荷時の調査手順

CPUにボトルネックがあることが確認されたとき、どのように対処すべきでしょうか。



①Performance Analyzerで、該当インスタンス名と日付を開いて、「CPU」タブから24時間トレンドを確認します。

→ [10:30 ～ 12:00]の時間帯でCPU使用率が100%に達しています。

②「OS指標」タブからCPU使用率構成の詳細を確認します。

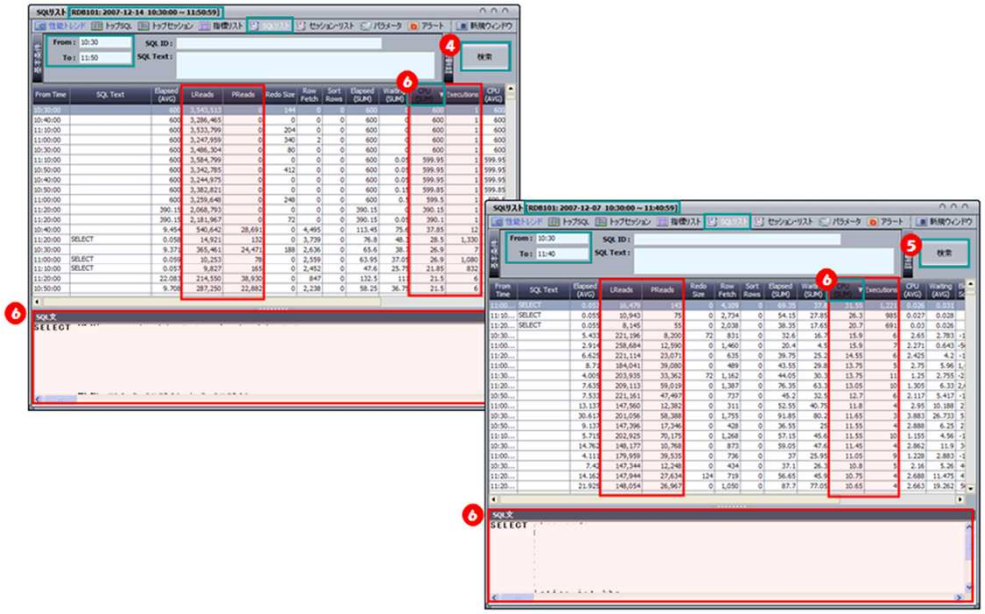
→「User CPU」が殆どを占めています

③「プロセス」タブから上位プロセスを確認します。

→Oracle関連ユーザーサーバープロセスが上位にランクしています。

→②と③から、Oracleデータベースのユーザー処理の過負荷がCPUリソースの枯渇現象の原因と考えられます。

活用TIPs：CPU過負荷時の調査手順



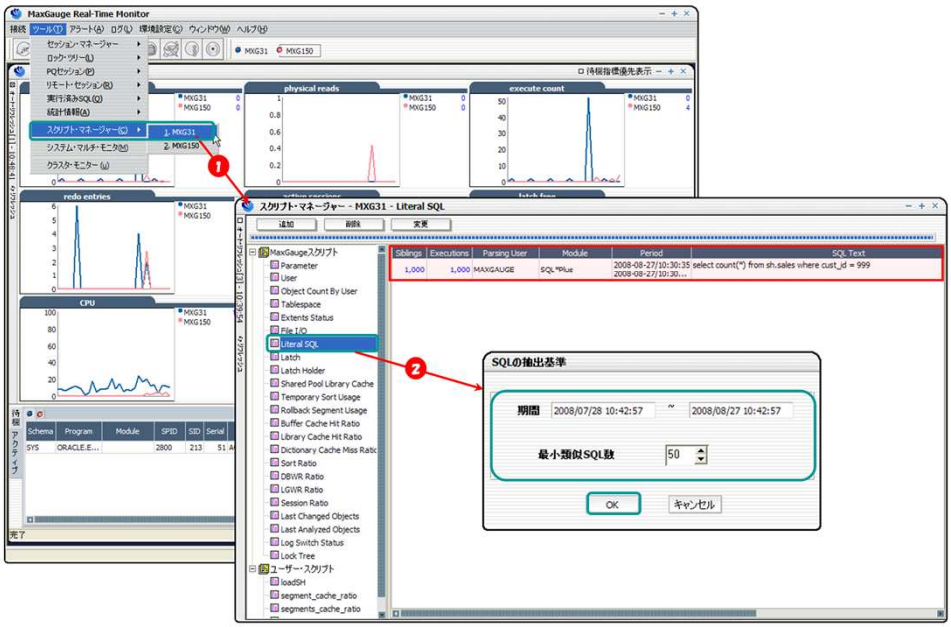
④CPU過負荷時の「SQLリスト」で、過負荷時間帯を指定し、「検索」押しでSQLを抽出します。

⑤通常時の「SQLリスト」で、「④」と同じ時間を指定し、「検索」押しでSQLを抽出します。

⑥「CPU(SUM)」項目の逆順にソートし、上位SQLの論理読取(LReads)、物理読取(PReads)、
CPU使用時間(CPU(SUM))、実行回数(Executions)の実行統計と、SQL文の詳細の違いを確認します。

→ 通常時(12/7)と比べて、CPU過負荷時(12/14)に上位SQL「SELECT MIN…」の論理読取が10倍以上増えています。
当時のアクセス対象のデータ量若しくは実行計画の変異を確認します。

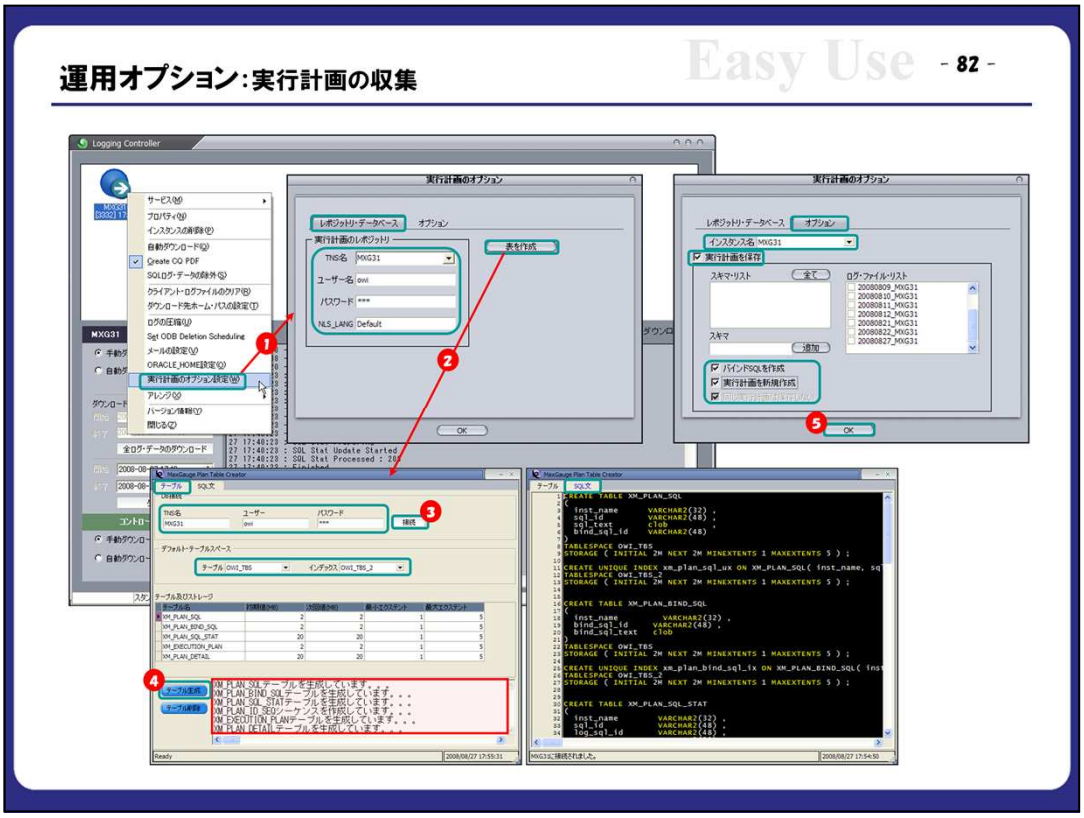
活用TIPs:「ORA-4031」対処 → リテラルSQL確認



①Real-Time Monitorの「スクリプト・マネージャ」メニューから該当インスタンスを選択します。

②「Literal SQL」をダブルクリックして、適切なSQL抽出の条件「期間、最小類似SQL数」を指定し、「OK」を押します。

→ 例の画面は、類似のSQL「1000」個が1回ずつ実行された結果を表示しています。



①Logging Controllerの対象インスタンスアイコンの右クリックメニューで「実行計画オプション」画面を開きます。

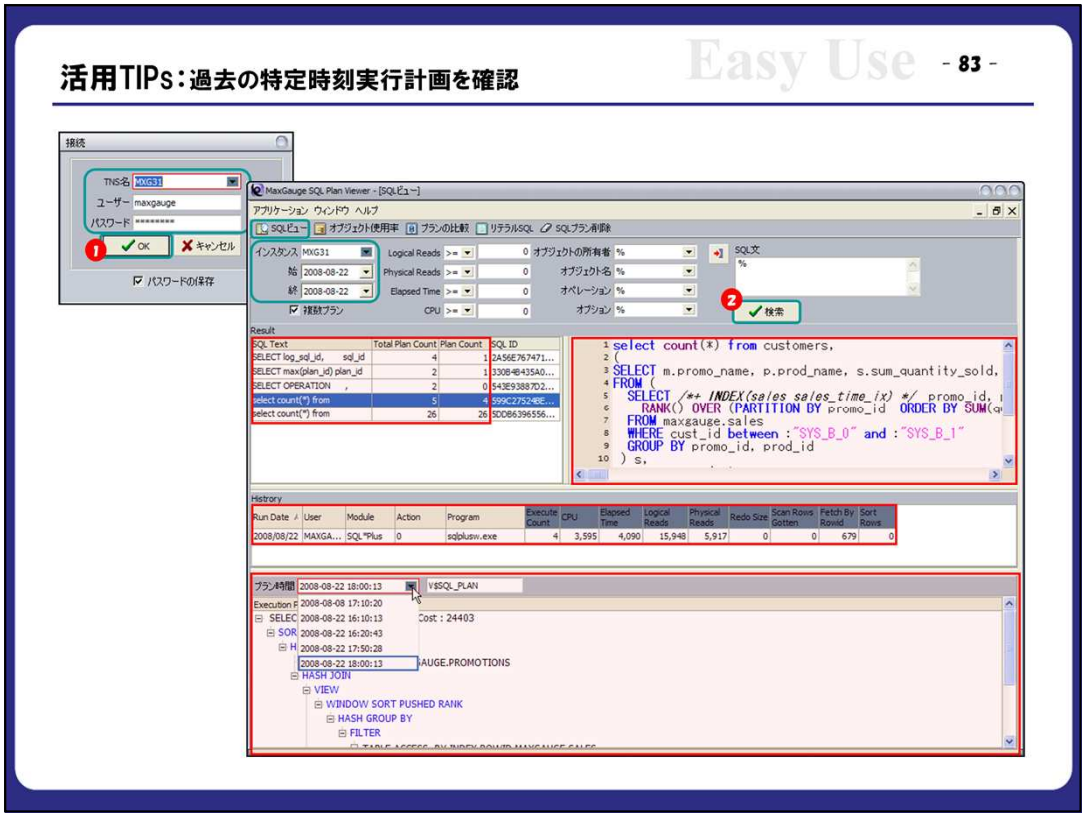
②「レジストリ・データベース」タブで、実行計画を保存するデータベースへの接続情報（TNS名、ユーザー名、パスワード）を設定し、「表を作成」を押します。

③「テーブル」タブで、実行計画を保存する表を作成するデータベースへの接続情報（TNS名、ユーザー名、パスワード）を設定し以下の情報を「接続」を行います。

④デフォルト「テーブル」と「インデックス」を指定し、「テーブル作成」を行います。
→ 表の作成時に詳細設定変更が必要な場合、「SQL文」タブのスクリプトを修正し、直接実行します。

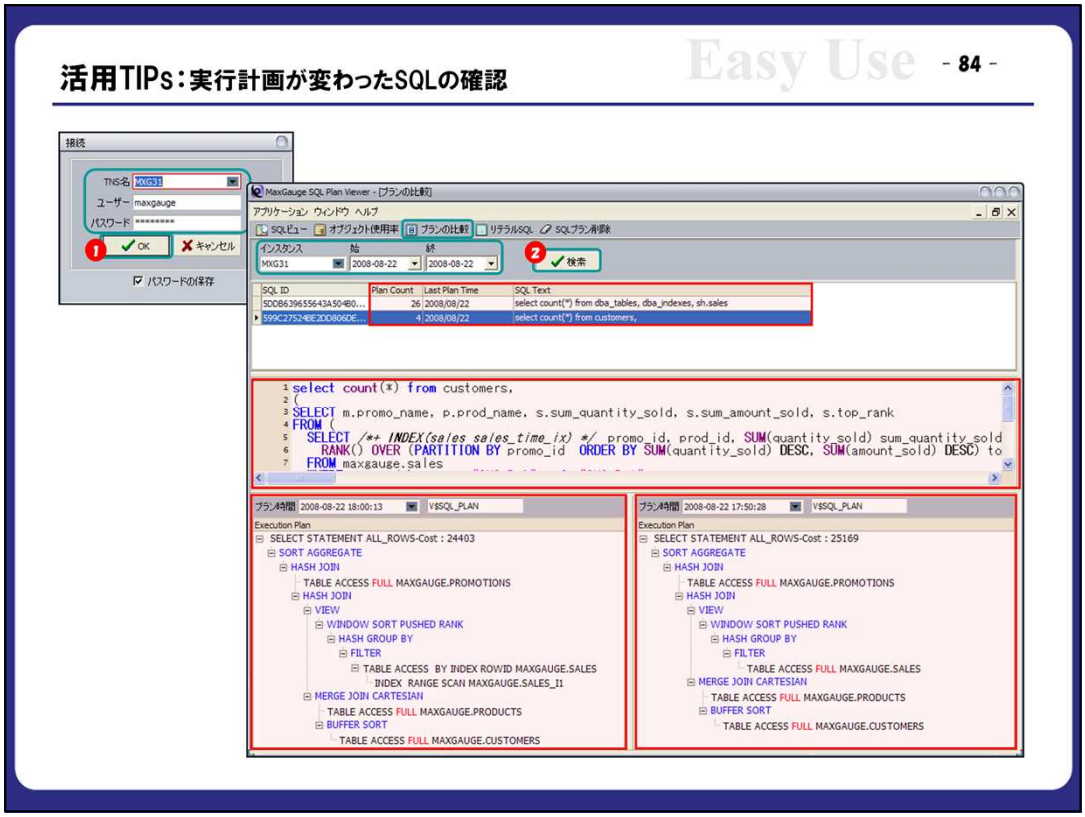
⑤「オプション」タブで、対象インスタンス名を選択して、「実行計画を保存」、「バインドSQLを作成」、「実行計画を新規作成」にチェックを外し、「OK」を押します。

※注意点：実行計画の収集範囲
MaxGaugeで収集済みSQLのみが、実行計画収集の対象になります。



①SQL PLAN Viewerを起動して、接続情報（TNS名、ユーザー、パスワード）を入力し、実行計画の格納データベースへ接続します。

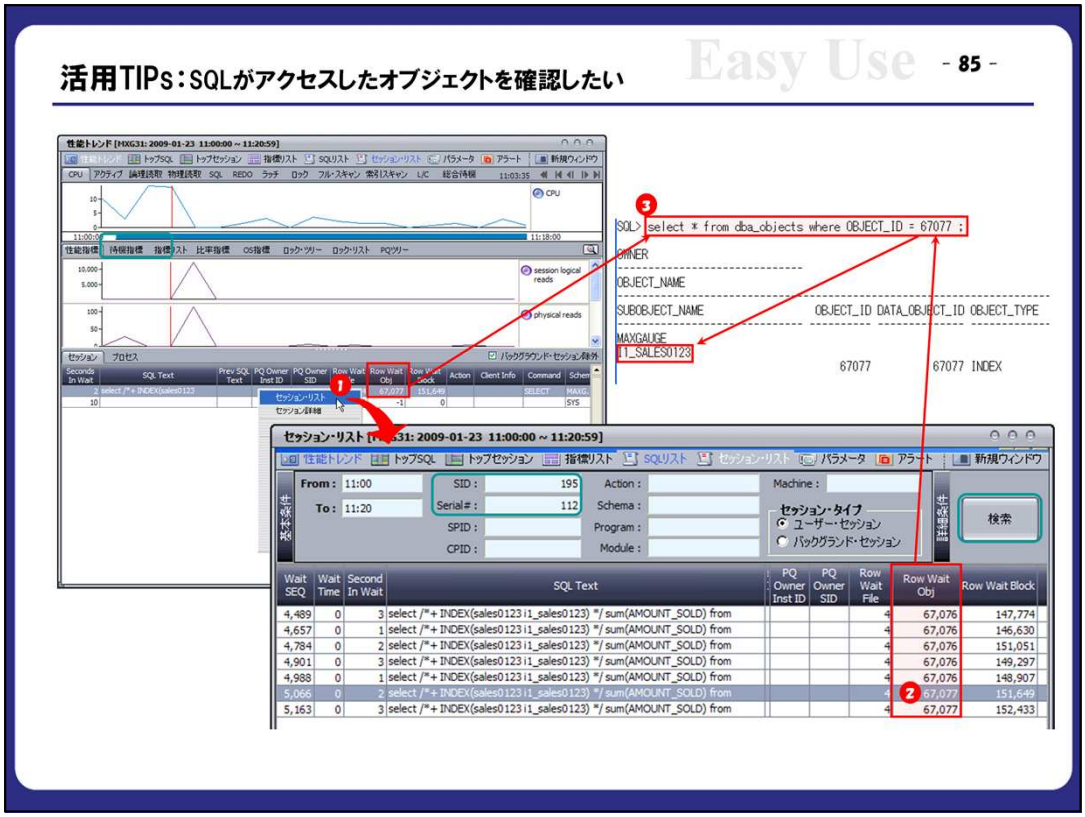
②「SQLビュー」で、該当の「インスタンス名」、対象期間「始、終」を指定して、「検索」を行います。
→ 各SQLの実行計画数を確認して、複数の場合収集時の詳細を確認します。



①SQL PLAN Viewerを起動して、接続情報（TNS名、ユーザー、パスワード）を入力し、実行計画の格納データベースへ接続します。

②「プランの比較」で、該当の「インスタンス名」、対象期間「始、終」を指定して、「検索」を行います。

→ 収集タイミングを選択し、左右の実行計画を比較・確認します。



①SQL及びセッションを特定して、セッションリストで該当セッションの全リストを検索します。
「Row Wait Obj」でこのタイミング(例では11:03:35)でアクセスしているオブジェクトを確認できます→③に移動。

②セッション情報の「Row Wait Obj」を確認します。
ここではSQLがアクセスしている表及び索引のオブジェクト番号が表示されます。

③「dba_objects」ビューより、「②」のオブジェクト名を確認します。